

Перспективи розвитку зварювальних інверторів дуже великі. Інверторні схеми відкривають нову сторінку у світі зварювання. В сьогодення, на їх принципі вже серйозно виготовляються багатофункційні зварювальні апарати. Найбільше розповсюдження отримали інвертори, які суміщають зварювання MMA, TIG і CUT або MIG/MAG, TIG і MMA. Зустрічаються і іннакші поєднання. Справа в тому, що інверторна схема дозволяє, так би мовити, “на ходу” змінювати тип зовнішніх вольт-амперних характеристик джерела живлення. Саме тип вольт-амперних характеристик є основною властивістю джерела для певного виду зварювання. І якщо звичайний апарат призначений, наприклад, для зварювання MIG/MAG, то штучним електродом він не має можливості зварювати. А інверторна схема – інша справа. Тут і тип характеристик(вольт-амперних) та інші параметри без проблем переналаштовуються під потрібний тип зварювання.

Список літератури

1. В.С. Милютин, М.П. Шалимов, С.М. Шанчуров. Источники питания для сварки. Москва, 2007
2. <http://www.welder.by/>

Одержано 16.98.10

УДК 699.841.01

О.В. Коваленко ас., В.В. Яцун ас., О.В. Яцун

Кіровоградський національний технічний університет

Розширення можливостей Autocad Architecture за допомогою платформи .Net Framework

У статті коротко описується основи програмування на платформі .Net Framework, у програмному середовищі AutoCAD Architecture фірми Autodesk, пояснюється структура й попереднє налагодження проекту у Microsoft Visual Studio, описується та аналізується концепція доступу до об'єктів, приводяться приклади керування вікном програми.

Інтерфейс прикладного програмування AutoCAD Architecture, програмування AutoCAD Architecture в середовищі Microsoft Visual Studio

Вступ. Autocad Architecture фірми Autodesk [1-3] це безпрецедентна гнучкість і свобода втілення архітектурних задумів, найбагатший набір готових будівельних об'єктів, таких як стіни, вікна, двері й багато чого іншого, а також провідні технології підготовки проектної документації від світового лідера CaprPішень, компанії Autodesk.

Звичний і добре знайомий по системах автоматизованого проектування Autocad [4] графічний інтерфейс гарантує 100% реалізацію вже наявних навичок машинного проектування при реалізації архітектурних проектів, а вбудована система тривимірного моделювання Autodesk VIZ Render дає можливість швидко і наочно презентувати задуми всім зацікавленим особам. Інтелектуальні засоби роботи з об'єктами на основі моделей дозволяють значно краще скоординувати роботу всіх учасників проекту, а сотні додаткових додатків, розроблених партнерами Autodesk, відкривають безмежні можливості для настроювання й доробки системи Autodesk Autocad Architecture (рис. 1) з урахуванням специфіки й характеру розв'язуваних завдань.

Але існує ряд специфічних завдань, з якими рано чи пізно зустрічається користувач

програми для реалізації яких необхідно розширювати можливості стандартної оболонки Autocad Architecture. Для цих цілей фірма Autocad створила інтерфейс прикладного програмування на основі платформи .Net Framework (далі «.Net API Autocad Architecture»).

Але на жаль на даний момент відсутня достовірна й практична документація із прикладами використання «.Net API Autocad Architecture» на українській або російській мові, що зводить нанівець можливість реалізувати деякі складні завдання. Дана стаття розглядає особливості використання «.Net API Autocad Architecture» очима програміста із прикладами програмування в середовищі Microsoft Visual Studio 2008 (і вище) мовою C#.

Структура й попереднє настроювання проекту Microsoft Visual Studio.

«.Net API Autocad Architecture» зібраний з різних Dll-Файлів, які містять широкий ряд класів, структур, методів і подій, що забезпечують доступ до об'єктів файлу креслення в додатках Autocad. Кожний Dll-Файл визначає різні простори імен, які використовуються для організації розміщення компонентів бібліотек зібраних згідно своєму функціоналу.

Існують три основні Dll-Файли «.Net API Autocad Architecture», які часто використовуються:

- Acdbmgd.dll. Використовується для роботи з об'єктами файлу креслення;
- Acmgd.dll. Використовується для роботи з додатком Autocad;
- Accui.dll. Використовується для роботи з файлами користувацьких налаштувань.

Перш ніж почати використовувати класи, структури, методи й події «.Net API Autocad Architecture», слід в проекті Microsoft Visual Studio обраної мови програмування попередньо підключити відповідні Dll-Файли, після чого встановити посилання на необхідні простори імен. Після вказівки в проекті посилання на Dll-Файл, що дозволяє використовувати «.Net API Autocad Architecture», необхідно встановити властивість «Copy Local» (локальна копія) бібліотеки, що підключається, у значення «False». Дана властивість відповідає за те, чи буде MS Visual Studio створювати копію Dll-Файлу, на який посилається, розміщуючи його в деякій директорії, що входить до складу зборки проекту, коли він буде компілюватися в Сіл-Код. Бібліотеки DLL «.Net API Autocad Architecture» можуть бути завантажені із сайту розроблювачів компанії Autodesk (<http://www.autodesk.com/adn>).

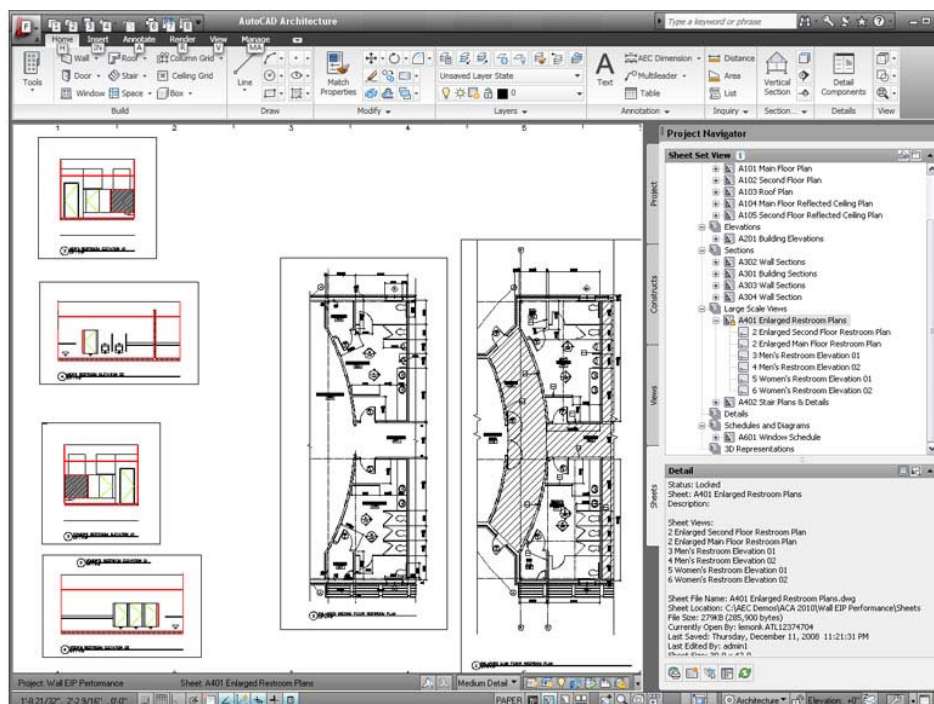


Рисунок 1 – Головне вікно програми Autocad Architecture

Microsoft Visual Studio є об'єктно-орієнтованим середовищем розробки, який працює незалежно від Autocad. Так як Microsoft Visual Studio є зовнішнім додатком, то він може взаємодіяти з Autocad (і з іншими додатками) як за допомогою .NET API, так і за допомогою бібліотек Activex/COM.

MS Visual Studio доступний у різних локалізаціях і варіантах публікацій. Для повноцінної роботи з «.Net API Autocad Architecture» на персональному комп'ютері необхідно встановити Microsoft Visual Studio 2008 SP1 і Microsoft .NET Framework 3.5 SP1. Для одержання доступу до об'єктів Autocad із проекту, створеного в Microsoft Visual Studio, необхідно додати в нього посилання на бібліотеки типів Autocad 2010 acax18enu.tlb, Autocad/Objectdbx Common 18.0, axdb18enu.tlb. Після додавання посилання на бібліотеку можна створювати в Microsoft Visual Studio різні змінні, що є типами Autocad, як приклад:

Код C#

```
Autodesk.Autocad.Interop.Acadapplication objacapp;  
Autodesk.Autocad.Interop.Common.Acadline objline;
```

Концепція доступу до об'єктів. Розглянемо концепції доступу до об'єктів Autocad за допомогою API керованого коду. «.Net API Autocad Architecture» дозволяє автоматично виконувати такі дії, як створення й зміна об'єктів, що зберігаються в базі даних файлу креслення, або зміна вмісту налаштувань файлу.

Об'єкт є основним будівельним блоком в Autocad .NET API. Кожний відображуваний об'єкт являє собою певну частину Autocad. В Autocad є безліч різних типів об'єктів. Деякі з них представлені в «.Net API Autocad Architecture»:

- графічні об'єкти (відрізки, окружності, текст, розміри й.т.буд.);
- налаштування стилів (стили шарів, типів ліній, розмірів і.т.буд.);
- організаційні структури (шари, групи, блоки й.т.буд.);
- відображення креслення (види, видові екрани й.т.буд.);
- креслення й сам додаток Autocad.

Об'єкти структуровані в ієрархічній формі з об'єкта додатка Autocad Application (основа структури). Ця структура часто розглядається як об'єктна модель (Object Model). На рисунку 2, а демонструється основна лінія зв'язку між об'єктом додатка Autocad і примітивами, що зберігаються в Blocktablerecord (блок табличних записів), такими як простір моделі (Model space).

Об'єкт Application (додаток, рис. 2, б) є кореневим об'єктом «.Net API Autocad Architecture». З нього можна одержати доступ до основного вікна додатка, а також до кожного відкритого файлу креслення. Одержавши посилання на креслення, програміст одержує доступ до об'єктів у його составі.

Як приклад об'єкт Application має властивість Documentmanager, яке повертає об'єкт Documentcollection (колекцію документів). Цей об'єкт забезпечує доступ до креслень, які відкриті в поточному сеансі роботи Autocad, а так само дозволяє створювати, зберігати й відкривати файли креслень. Інші властивості об'єкта Application забезпечують доступ до специфічних для додатка властивостей, таким як Infocenter (центр інформації), основне вікно додатка й панель статусу. Властивість Mainwindow дозволяє одержати доступ до імені додатка, розміру основного вікна, його розташуванню на екрані й видимості.

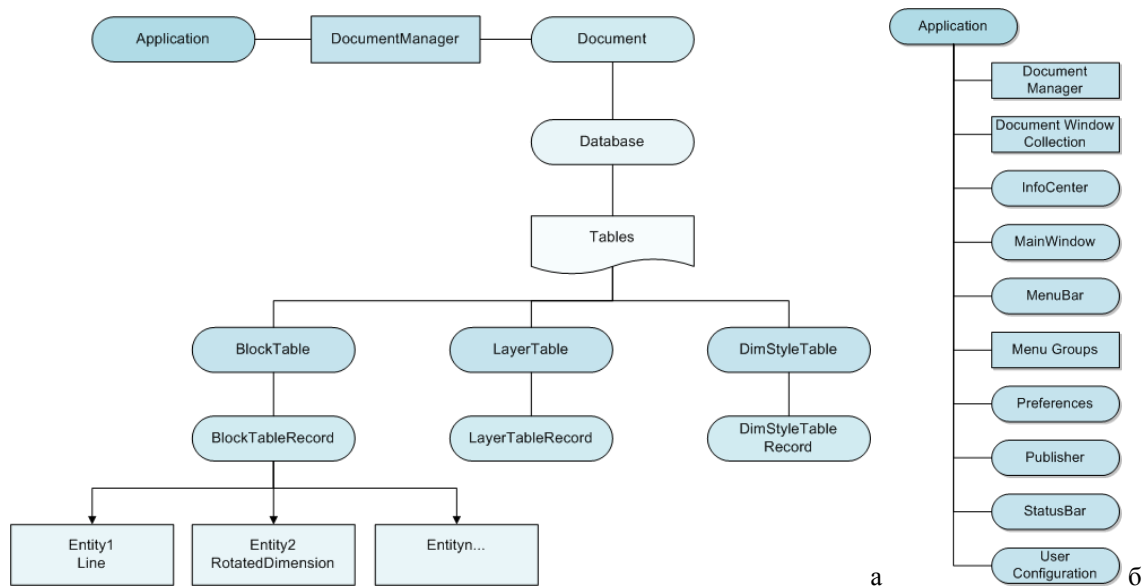


Рисунок 2 – Об'єкти структуровані в ієрархічній формі: а-а зв'язку між об'єктом додатка й примітивами; б-б кореневий об'єкт «.Net API Autocad Architecture»

У той час як більшість властивостей об'єкта *Application* дозволяють одержати доступ до об'єктів в «.Net API Autocad Architecture», доступ до деяких об'єктів можна одержати й за допомогою *Autocad Activex Automation*. Ці властивості включають *CompВерсію* об'єкта додатка (*Acadapplication*), панель меню (*Menubar*), завантажені групи меню (*Menugroups*) і переваги (*Preferences*).

З кореневого об'єкта *Application* можна одержати доступ до наступних об'єктів:

- *Documentmanager* (менеджер документів). Контейнер, що містить у собі об'єкти документів (усі відкриті креслення);
- *Documentwindowcollection* (колекція вікон документів). Контейнер для всіх вікон документів (вікна для кожного документа з колекції *Documentmanager*);
- *Infocenter* (інформаційний центр). Містить посилання на панель *Infocenter*;
- *Mainwindow* (основне вікно додатка). Містить посилання на основне вікно додатка *Autocad*;
- *Menubar* (панель меню). Містить посилання на ComОб'єкт *Menubar*, що представляє собою меню *Autocad*;
- *Menugroups* (групи меню). Містить посилання на ComОб'єкт *Menugroups* який, у свою чергу, містить найменування груп налаштувань для кожного завантаженого *CuіxфФайлу*;
- *Preferences* (користувачські налаштування). Містить довідкову інформацію COM object, яка дозволить змінювати налаштування, доступні в діалогові вікні *Options*;
- *Publisher* (публікація). Містить посилання на об'єкт *Publisher*, який використовується для публікації креслень;
- *Statusbar* (панель статусу). Містить посилання на об'єкт *Statusbar*, що є статусним рядком основного вікна додатка;
- *Userconfiguration* (користувачська конфігурація). Містить посилання на об'єкт *Userconfiguration*, що дозволяє працювати зі збереженими користувачськими профілями.

Керування вікном додатка (*Application Window*). Здатність керування основним вікном додатка (*Application window*) надає розроблювачам гнучкість, необхідну для створення ефективних і інтелектуальних додатків. Використовуючи методи й властивості, наявних в об'єкті додатка (*Application object*), можна змінити позицію, розмір і видимість вікна додатка. Так само можна використовувати властивість *Windowstate* (статус вікна) для мінімізації, максимізації й перевірки поточного статусу вікна додатка.

Позиція й розміри вікна додатка (Application window):

Код C#

```
using System.Drawing;
using Autodesk.AutoCAD.Runtime;
using Autodesk.AutoCAD.ApplicationServices;
[CommandMethod("Positionapplicationwindow")]
public static void Positionapplicationwindow()
{
    // Позиціонування вікна додатка
    Point ptapp = new Point(0, 0);
    Application.Mainwindow.Location = ptapp;
    // Призначення розміру вікна додатка
    Size szapp = new Size(400, 400);
    Application.Mainwindow.Size = szapp;
}
```

Мінімізація й максимізація вікна додатка Autocad:

Код C#

```
using System.Drawing;
using Autodesk.AutoCAD.Runtime;
using Autodesk.AutoCAD.ApplicationServices;
using Autodesk.AutoCAD.Windows;
[CommandMethod("Minmaxapplicationwindow")]
public static void Minmaxapplicationwindow()
{
    // Мінімізація вікна додатка
    Application.Mainwindow.Windowstate = Window.State.Minimized;
    System.Windows.Forms.MessageBox.Show("Мінімізоване", "Minmax",
        System.Windows.Forms.MessageBoxButtons.OK,
        System.Windows.Forms.MessageBoxIcon.None,

System.Windows.Forms.MessageBoxdefaultbutton.Button1,

System.Windows.Forms.MessageBoxoptions.Servicenotification);
    // Максимізація вікна додатка
    Application.Mainwindow.Windowstate = Window.State.Maximized;
    System.Windows.Forms.MessageBox.Show("Максимізоване",
"Minmax");
}
```

Визначення поточного стану вікна додатка:

Код C#

```
using Autodesk.AutoCAD.Runtime;
using Autodesk.AutoCAD.ApplicationServices;
using Autodesk.AutoCAD.Windows;
[CommandMethod("Currentwindowstate")]
public static void Currentwindowstate()
{
    System.Windows.Forms.MessageBox.Show("The application window
is " +
    Application.Mainwindow.Windowstate.ToString(), "Window
State");
}
```

Керування видимістю вікна додатка:

Код C#

```

using Autodesk.Autocad.Runtime;
using Autodesk.Autocad.ApplicationServices;
using Autodesk.Autocad.Windows;
[CommandMethod("Hidewindowstate")]
public static void Hidewindowstate()
{
    // Сховати вікно додатка
    Application.Mainwindow.Visible = false;
    System.Windows.Forms.MessageBox.Show("Невидимо",
"Показати/Сховати");
    // Відобразити вікно додатка
    Application.Mainwindow.Visible = true;
    System.Windows.Forms.MessageBox.Show("Видиме",
"Показати/Сховати");
}

```

Висновки. Autocad Architecture успадкував від Autocad усю його функціональність і, крім цього, доповнений спеціалізованими функціями для архітектурно-будівельного проектування й формування документації. Звичний спосіб роботи й інтерфейс Autocad дозволяють приступитися до роботи без яких-небудь утруднень і негайно відчутти вигравш у продуктивності.

Переваги одержувані від застосування «.Net API Autocad Architecture»:

- Відкритий програмний доступ до креслень Autocad Architecture з різних середовищ розробки додатків. Раніше програмісти були обмежені Activex і мовами, що підтримують COM, Autolisp і C++ з Objectarx;
- Інтеграція з іншими WindowsoДодатками, такими як MS Excel, і Word стає більш ефективною й легкою при використанні.NET API;
- Бібліотека .NET Framework розроблена як для 32-бітної, так і для 64-бітної версій операційної системи Windows. Що дозволяє підвищити швидкодію розрахунків.

Переваги одержувані від застосування середовища розробки Microsoft Visual Studio:

- мова високого рівня C# підвищує швидкість розробки програм;
- потужне й доступне середовище розробки, яке не вимагає багато часу на навчання в роботі з нею;
- візуальні, інтуїтивні й гнучкі засоби розробки діалогових вікон.

Microsoft Visual Studio дозволяє створювати різні варіанти рішень, оформлених як зовнішніх додатків і бібліотек, так і додатків, призначених для завантаження й використання їх у самому Autocad Architecture;

- на відміну від проектів інших середовищ розробок, проект, написаний на Microsoft Visual C# (.Net) не має втрати продуктивності при завантаженні його в 64-бітну версію Autocad Architecture.

Список літератури

1. 4D Design Solutions, LLC dba 4D Technologies. AutoCAD® Architecture 2010 Tutorial Series/ AutoCAD Architecture. 4D Design Solutions, LLC dba 4D Technologies 2010.
2. ASCENT. AutoCAD® Architecture 2010 Fundamentals/ AutoCAD Architecture. ASCENT. 2010.
3. ASCENT. AutoCAD® Architecture 2010 Advanced / AutoCAD Architecture. ASCENT 2010.
4. Алан Джефферис. AutoCAD 2004 для архитекторов// Алан Джефферис, Майкл Джоунз, Тереза Джефферис, 2006. – 864 с.

Одержано 16.08.10